

明 細 書

キャッシュメモリおよびその制御方法

技術分野

[0001] 本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリおよびその制御方法に関する。

背景技術

[0002] 近年、キャッシュメモリは、メインメモリへのアクセス時間を短縮しプロセッサの処理能力の向上を図るために広く用いられている。

[0003] プロセッサからキャッシュメモリにデータが書き込まれた場合、メインメモリとキャッシュメモリでデータの一貫性を確保するためにキャッシュメモリからメインメモリへ書き戻す必要がある。この書き戻しの制御方式には、ライトスルー方式およびライトバック方式がある。

[0004] ライトスルー方式では、プロセッサからキャッシュメモリへのライト命令実行時にキャッシュメモリと主記憶との両方に書き込みを行う。このライトスルー方式によれば、キャッシュメモリにデータを書き込む毎にライトバックペナルティが発生するので、メモリライトに対してメモリアクセス時間を短縮する効果が薄い。

[0005] 一方、ライトバック方式は、ライト命令実行時にキャッシュメモリのみに書き込みを行い、キャッシュミスによりデータラインを新たな別のデータに置換する直前にキャッシュメモリからメインメモリにライトバックする。このライトバック方式は、メモリライト毎にライトバックペナルティが発生するわけではないが、メモリリード及びメモリライトにおいてキャッシュミス発生時にはロードペナルティに加えてライトバックペナルティを発生させるためキャッシュミス時のペナルティが増えることになる。ここでロードペナルティは、メインメモリからキャッシュメモリに新たなデータをロードすることによるペナルティをいう。ライトバックペナルティは、キャッシュメモリからメインメモリにデータを書き戻すことによるペナルティをいう。

[0006] このようなキャッシュミスペナルティを低減する従来技術は、例えば、特許文献1等に開示されている。

[0007] この従来技術によれば、ライトバック方式のキャッシングメモリにおいて、キャッシングメモリからデータを追い出すためのメインメモリへの通常の書き戻し処理とは別に、キャッシングメモリのデータをメインメモリに書き戻すだけのための書き戻し処理を行う。後者の処理は、先ずキャッシングメモリの一つのエントリーのデータがデータであるか否かを判定し、データであった場合はそのデータをメインメモリに書き戻す。そして、書き戻しが正常終了し、かつキャッシングメモリ上のデータが変化していないことを確認し、キャッシング状態をクリーンとする。

[0008] このようにして従来技術におけるキャッシングメモリは、キャッシングメモリからメインメモリへのトラフィックを減少させ、システム全体の性能向上を図っている。

特許文献1:特開平6-309231号公報

発明の開示

発明が解決しようとする課題

[0009] しかしながら、上記従来技術によれば、キャッシングミスが発生する前にデータが書き込まれた場合に、却って効率が低下するという問題がある。

[0010] 上記課題に鑑み本発明は、キャッシングミス時のライトバックペナルティを効率よく低減するキャッシングメモリを提供することを目的とする。

課題を解決するための手段

[0011] 上記目的を達成するため本発明のキャッシングメモリは、ラインデータを保持するキャッシングエントリー毎に、当該キャッシングエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を付与する付与手段と、キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すデータフラグがセットされているキャッシングエントリーを選択する選択手段と、選択されたキャッシングエントリーのラインデータをキャッシングミスの発生と無関係にメモリへライトバックするライトバック手段とを備える。

[0012] この構成によれば、キャッシングを終了してよいキャッシングエントリーを、キャッシングミスが発生する前にライトバックするので、キャッシングミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシ

ュメモリの効率を向上させ、アクセス速度を向上させることができる。

- [0013] ここで、前記付与手段は、プロセッサから指定されたアドレス範囲を保持する保持手段と、保持手段に保持されたアドレス範囲に属するラインデータを保持するキャッシュエントリーを探索する探索手段と、探索されたキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定する設定手段とを備える構成としてもよい。
- [0014] ここで、前記探索手段は、保持手段に保持されたアドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第1変換手段と、保持手段に保持されたアドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第2変換手段と、前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段とを備える構成としてもよい。
- [0015] この構成によれば、プロセッサは、キャッシングを終了してよいことを示すキャッシング終了属性を設定すべきアドレス範囲として、キャッシュメモリのラインサイズやライン境界とは無関係に任意のアドレスを指定することができる。言い換れば、プログラマやコンパイラは、ラインサイズ及びライン境界を意識する必要がなくなるので、ライトバックするためのラインアドレスを管理をしなくてもよく、プログラムの作成を容易にことができる。
- [0016] ここで、前記キャッシュメモリは、さらに、キャッシング発生時に、キャッシングを終了してよいことを示すキャッシング終了属性が付されたキャッシュエントリーをリプレース対象として選択する選択手段を備える構成としてもよい。
- [0017] この構成によれば、アクセス頻度の低いデータを保持するキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定することにより、当該キャッシュエントリーは真っ先にリプレースされるので、アクセス頻度の低いデータがキャッシュメモリに居座ることによるキャッシングの誘発を低減することができる。
- [0018] ここで、前記付与手段は、プロセッサにおいてキャッシングを終了してよいことを示

すキャッシング終了属性の付与とデータを書き込みとを命令内容とするストア命令が実行されたことを検出する命令検出手段と、当該命令によって書き込みがなされたキャッシュエントリーに、当該キャッシング終了属性を設定する設定手段とを備える構成としてもよい。

- [0019] この構成によれば、キャッシング終了属性の付与とデータを書き込みとを命令するストア命令により、キャッシングを終了してよいことを示すキャッシング終了を簡単に設定することができる。
- [0020] ここで、前記ライトバック手段は、メモリバスが空きサイクルになっているとき、キャッシングエントリーのデータをメモリへライトバックする構成としてもよい。
- [0021] この構成によれば、空きサイクルを利用した効率のよいライトバックを行うことができる。
- [0022] ここで、前記キャッシングエントリーは、1ラインを構成する複数のサブライン毎にダーティフラグを有し、前記ライトバック手段は、選択手段に選択されたキャッシングエントリーについてダーティなサブラインのみをライトバックする構成としてもよい。
- [0023] この構成によれば、ダーティなサブラインのみをライトバックするので、ライトバックを効率よく行うことができる。

また、本発明のキャッシングメモリの制御方法についても上記と同様の手段、作用を有する。

発明の効果

- [0024] 上記のように、本発明のキャッシングメモリによれば、以後に書き込みが行われないであろうキャッシングエントリーにキャッシングを終了してよいことを示すキャッシング終了属性を付与し、当該キャッシングエントリーを、キャッシングミスの発生とは無関係につまりキャッシングミスが発生する前にライトバックすることができる。その結果、キャッシングミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシングメモリの効率を向上させ、アクセス速度を向上させることができる。
- [0025] プロセッサは、ライトバックのためのラインアドレスの管理をする必要がなく、キャッシングメモリのラインサイズ及びライン境界とは無関係に任意のアドレス範囲を指定するこ

とができる。

図面の簡単な説明

[0026] [図1]本発明の実施の形態1におけるプロセッサ、キャッシングメモリ、メモリを含むシステムの概略構成を示すブロック図である。

[図2]キャッシングメモリの構成例を示すブロック図である。

[図3]キャッシングエントリーの詳細なビット構成を示す図である。

[図4]制御部の構成を示すブロック図である。

[図5]Cフラグ設定部の構成例を示すブロック図である。

[図6A]スタートアドレスレジスタにスタートアドレスを書き込む命令の一例を示す。

[図6B]サイズレジスタにサイズを書き込む命令の一例を示す。

[図6C]コマンドレジスタにコマンドを書き込む命令の一例を示す。

[図6D]コマンドの一例を示す。

[図7]スタートアライナ及びエンドアライナの説明図を示す。

[図8]フラグ書換部におけるCフラグ設定処理の一例を示すフローチャートである。

[図9]クリーニング処理部におけるクリーニング処理の一例を示すフローチャートである。

[図10]本発明の実施の形態2におけるキャッシングメモリの構成を示すブロック図である。

。

[図11]キャッシングエントリーのビット構成を示す図である。

[図12]制御部の構成を示すブロック図である。

[図13]リプレース部による使用フラグの更新例を示す。

[図14A]ウイークフラグが存在しない場合にキャッシングエントリーがリプレースされる様子を示す図である。

[図14B]リプレース処理におけるウイークフラグWの役割を示す説明図である。

[図15]Wフラグ設定部におけるWフラグ設定処理を示すフローチャートである。

[図16]クリーニング処理部におけるクリーニング処理を示すフローチャートである。

[図17]リプレース部におけるUフラグ更新処理を示すフローチャートである。

[図18]リプレース部におけるリプレース処理を示すフローチャートである。

符号の説明

[0027] 1 プロセッサ
2 メモリ
3 キャッシュメモリ
20 アドレスレジスタ
21 メモリI/F
30 デコーダ
31a～31d ウェイ
32a～32d 比較器
33a～33d アンド回路
34 オア回路
35 セレクタ
36 セレクタ
37 デマルチプレクサ
38 制御部
39 クリーニング処理部
40 Cフラグ設定部
41 リプレース部
131b～131d ウェイ
138 制御部
139 クリーニング処理部
140 Wフラグ設定部
401 コマンドレジスタ
402 スタートアドレスレジスタ
403 サイズレジスタ
404 加算器
405 スタートアライナ
406 エンドアライナ

407 フラグ書換部

発明を実施するための最良の形態

[0028] (実施の形態1)

<全体構成>

図1は、本発明の実施の形態1におけるプロセッサ1、キャッシングメモリ3、メモリ2を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシングメモリ3は、プロセッサ1およびメモリ2を有するシステムに備えられる。

[0029] キャッシュメモリ3は、キャッシングの単位データ(ラインデータと呼ぶ)を保持するキャッシングエントリー毎に、当該キャッシングエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を保持し、キャッシングミスの発生の有無に関わらず、キャッシングを終了してよいことを示すキャッシング終了属性を有しつつプロセッサから書き込みされたことを示すダーティフラグがセットされているキャッシングエントリーを選択し、選択されたキャッシングエントリーのデータをメモリへライトバックするよう構成されている。ここで、キャッシングを終了してよいことを示すキャッシング終了属性は、例えば、以後に書き込みが行われないであろうキャッシングエントリーや、以降に読み書きされないであろうキャッシングエントリー等に付与される。

<キャッシングメモリの構成>

[0030] 以下、キャッシングメモリ3の具体例として、4ウェイ・セット・アソシエイティブ方式のキャッシングメモリに本発明を適用した場合の構成について説明する。

図2は、キャッシングメモリ3の構成例を示すブロック図である。同図のように、キャッシングメモリ3は、アドレスレジスタ20、メモリ1/F21、デコーダ30、4つのウェイ31a～31d(以下ウェイ0～3と略す)、4つの比較器32a～32d、4つのアンド回路33a～33d、オア回路34、セレクタ35、36、デマルチプレクサ37、制御部38を備える。

[0031] アドレスレジスタ20は、メモリ2へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは32ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、21ビットのタグアドレス、4ビットのセットインデックス(図中のSI)、5ビットのワードインデックス(図中のWI)を含む。ここで、タグアドレスはウェイにマッピングされるメモリ中の領域(そのサイズはセット数×ブロックである)を指す。この領

域のサイズは、タグアドレスよりも下位のアドレスビット(A10～A0)で定まるサイズつまり2kバイトであり、1つのウェイのサイズでもある。セットインデックス(SI)はウェイ0～3に跨る複数セットの1つを指す。このセット数は、セットインデックスが4ビットなので16セットある。タグアドレスおよびセットインデックスで特定されるキャッシングエントリーは、リプレース単位であり、キャッシングメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり128バイトである。1ワードを4バイトとすると、1ラインデータは32ワードである。ワードインデックス(WI)は、ラインデータを構成する複数ワード中の1ワードを指す。アドレスレジスタ20中の最下位2ビット(A1、A0)は、ワードアクセス時には無視される。

- [0032] メモリI/F21は、キャッシングメモリ3からメモリ2へのデータのライトバックや、メモリ2からキャッシングメモリ3へのデータのロード等、キャッシングメモリ3からメモリ2をアクセスするためのI/Fである。
- [0033] デコーダ30は、セットインデックスの4ビットをデコードし、4つのウェイ0～3に跨る16セット中の1つを選択する。
- [0034] 4つのウェイ0～3は、同じ構成を有数する4つのウェイであり、4×2kバイトの容量を有する。各ウェイは、16個のキャッシングエントリーを有する。
- [0035] 図3に1つのキャッシングエントリーにおける詳細なビット構成を示す。同図のように、1つのキャッシングエントリーは、バリッドフラグV0～V3、21ビットのタグ、128バイトのラインデータ、キャッシング終了属性(図中のCフラグ)、ダーティフラグD0～D3を有する。
- [0036] タグは21ビットのタグアドレスのコピーである。
ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の128バイトデータのコピーであり、32バイトの4つのサブラインからなる。
- [0037] バリッドフラグV0～V3は、4つのサブラインに対応し、サブラインが有効か否かを示す。
- [0038] キャッシング終了属性(クリーニングフラグC)は、キャッシングを終了してよいかどうかを示す。例えば、当該キャッシングエントリーに以降に書き込みが行われないかどうか

を意味する。C=0は、以降に書き込みがなされる可能性があることを意味する。C=1は、以降に書き込みがなされないことを意味し、ダーティであればクリーニング(ライトバック)すべきであることを意味する。

- [0039] ダーティフラグD0ーD3は、4つのサブラインに対応し、そのサブラインにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。
- [0040] 比較器32aは、アドレスレジスタ20中のタグアドレスと、セットインデックスにより選択されたセットに含まれる4つのタグ中のウェイ0のタグとが一致するか否かを比較する。比較器32bー32cについても、ウェイ31bー31dに対応すること以外は同様である。
- [0041] アンド回路33aは、バリッドフラグと比較器32aの比較結果とが一致するか否かを比較する。この比較結果をh0とする。比較結果h0が1である場合は、アドレスレジスタ20中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ0においてヒットしたことを意味する。比較結果h0が0である場合は、ミスヒットしたことを意味する。アンド回路33bー33dについても、ウェイ31bー31dに対応すること以外は同様である。その比較結果h1ーh3は、ウェイ1ー3でヒットしたかミスしたかを意味する。
- [0042] オア回路34は、比較結果h0ーh3のオアをとる。このオアの結果をhitとする。hitは、キャッシュメモリにヒットしたか否かを示す。
- [0043] セレクタ35は、選択されたセットにおけるウェイ0ー3のラインデータのうち、ヒットしたウェイのラインデータを選択する。
- [0044] セレクタ36は、セレクタ35により選択された32ワードのラインデータにうち、ワードインデックスに示される1ワードを選択する。
- [0045] デマルチプレクサ37は、キャッシュエントリにデータを書き込む際に、ウェイ0ー3の1つに書き込みデータを出力する。この書き込みデータはワード単位でよい。
- [0046] 制御部38は、キャッシュメモリ3の全体の制御を行う。特に、Cフラグの設定とCフラグに従うクリーニング(ライトバック)とを行う。
＜制御部の構成＞

[0047] 図4は、制御部38の構成を示すブロック図である。同図のように、制御部38は、クリーニング処理部39とCフラグ設定部40とを含む。

クリーニング処理部39は、C=1が設定されているキャッシュエントリーを探索し、当該キャッシュエントリーがデータイであればライトバックする。

[0048] Cフラグ設定部40は、プロセッサ1からのコマンドに応じてクリーニングフラグCを設定する。プロセッサ1は、もはや書き込みをしないキャッシュエントリーについてクリーニングフラグの設定を指示するコマンドをキャッシュメモリ3に対して発行する。

<Cフラグ設定部の構成>

[0049] 図5は、Cフラグ設定部40の構成例を示すブロック図である。同図のようにCフラグ設定部40は、コマンドレジスタ401、スタートアドレスレジスタ402、サイズレジスタ403、加算器404、スタートアライナ405、エンドアライナ406、フラグ書換部407を備える。

コマンドレジスタ401は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたCフラグ設定コマンドを保持する。図6(c)に、コマンドレジスタ401にコマンドを書き込む命令の一例を示す。この命令は、通常の転送命令(mov命令)であり、ソースオペランドとしてコマンドを、デスティネーションオペランドとしてコマンドレジスタ(CR)401を指定している。図6(d)に、コマンドの一例を示す。このコマンドは、Cフラグ設定コマンドを示す特定のコードである。Cフラグ設定コマンドは、スタートアドレスレジスタ402に保持されたスタートアドレスからサイズレジスタ403に保持されたサイズのアドレス範囲に対応するデータを保持するキャッシュエントリーに対して、Cフラグを設定することを指示するコマンドである。

[0050] スタートアドレスレジスタ402は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたスタートアドレスを保持する。このスタートアドレスはCフラグを設定すべきアドレス範囲の開始位置を示す。図6(a)に、スタートアドレスレジスタ402にスタートアドレスを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。

[0051] サイズレジスタ403は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたサイズを保持する。このサイズは、スタートアドレスからのアドレ

ス範囲を示す。図6(b)に、サイズレジスタ403にサイズを書き込む命令の一例を示す。この命令も、図6(c)と同様に通常の転送命令(mov命令)である。なお、サイズの単位は、バイト数であっても、ライン数(キャッシュエントリー数)であってもよく、予め定められた単位であればよい。

- [0052] 加算器404は、スタートアドレスレジスタ402に保持されたスタートアドレスとサイズレジスタ403に保持されたサイズとを加算する。加算結果は、アドレス範囲の終了位置を指すエンドアドレスである。加算器404は、サイズがバイト数指定の場合はバイトアドレスとして加算し、サイズがライン数指定の場合はラインアドレスとして加算すればよい。
- [0053] スタートアライナ405は、スタートアドレスをライン境界の位置に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意のアドレスをスタートアドレスとして指定することができる。
- [0054] エンドアライナ406は、エンドアドレスをライン境界の位置に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意の大きさを上記サイズとして指定することができる。
- [0055] 図7に、スタートアライナ405及びエンドアライナ406の説明図を示す。同図において、プロセッサ1から指定されたスタートアドレスはラインNの途中の任意の位置を指す。スタートアライナ405は、次のライン(N+1)の先頭を指すよう調整し、調整後のアドレスをラインスタートアドレスとして出力する。ラインスタートアドレスが指すラインをスタートラインと呼ぶ。
- [0056] また、エンドアドレスはラインMの途中の任意の位置を指す。エンドアライナ406は、直前のライン(M-1)の先頭を指すよう調整し、調整後のアドレスをラインエンドアドレスとして出力する。ラインエンドアドレスが指すラインをエンドラインと呼ぶ。
- [0057] この場合、スタートライン(ライン(N+1))からエンドライン(ライン(M-1))までの各ライン(キャッシュエントリー)にCフラグが設定されることになる。このように、スタートアライナ405及びエンドアライナ406がプロセッサ1から指定されたスタートアドレスからエンドアドレスまでのアドレス範囲よりも内側にラインしているのは、ラインNとラインMの外側の部分にはプロセッサ1から書き込みが発生する可能性があるからである。

[0058] フラグ書換部407は、ラインスタートアドレスが指すラインからラインエンドアドレスが指すラインまで(図7の例ではライン(N+1)からライン(M-1)まで)、キャッシュメモリ3にエントリーされていればCフラグを1に設定する。

<Cフラグ設定処理>

[0059] 図8は、フラグ書換部407におけるCフラグ設定処理の一例を示すフローチャートである。

フラグ書換部407は、コマンドレジスタ401にCフラグ設定コマンドが保持されている場合、スタートラインからエンドラインまでの各ラインアドレスを順に出力しながらループ1の処理(S82～S86)を行う。フラグ書換部407は、各ラインについて同じ処理を行うので、ここでは1ライン分の処理について説明する。

[0060] すなわち、フラグ書換部407は、キャッシュメモリ3がプロセッサ1からアクセスされていない間に、ラインアドレスをアドレスレジスタ20に出力し(S83)、アドレスレジスタ20のタグアドレスとキャッシュエントリーのタグとを比較器32a～32dに比較させ、ヒットするかどうかを判定する(S84)。さらにフラグ書換部407は、ヒットした場合には、ヒットしたキャッシュエントリーに対してCフラグを1にセットし(S85)、ミスヒットした場合には、キャッシュメモリにエントリーされていないのでなにもしない。

[0061] これにより、スタートラインからエンドラインまでの各ラインについて、キャッシュメモリ3にエントリーされている場合には、Cフラグに1が設定される。

<クリーニング処理>

[0062] 図9は、クリーニング処理部39におけるクリーニング処理の一例を示すフローチャートである。

同図のように、クリーニング処理部39は、ループ1の処理(S900～S913)において、セットインデックス(SI)0～15を順に指定する(S901)ことにより、16個の全てのセットに対してループ2の処理を行う。ループ2の処理(S900～S913)において、クリーニング処理部39は、セット内のウェイ0～3のCフラグを読み出す(S903)ことにより、C=1のキャッシュエントリーを探索する(S904)。ループ3の処理(S905～910)において、クリーニング処理部39は、C=1のキャッシュエントリーに対して、サブライン単位のダーティフラグを読み出し(S906)、ダーティであれば(S907)、そのサブラン

のデータをメモリ2に書き戻し(S908)、当該ダーティフラグを0にリセットする(S909)。このサブラインデータの書き戻しにおいて、クリーニング処理部39は、ループ4の処理(S920～S923)のように、空きサイクルにおいて(S920)、1ワードずつ書き戻す(S922)。

- [0063] このように、クリーニング処理部39は、全てのキャッシュエントリーのCフラグを順にチェックして、C=1のキャッシュエントリーを探索し、ダーティであればキャッシュメモリ3からメモリ2に書き戻す。
- [0064] 以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、プロセッサ1から指定されたアドレス範囲に属するキャッシュエントリーに対して、キャッシングを終了してよいかどうかを示すキャッシング終了属性(Cフラグ)を付与し、これ以上書き込みされないキャッシュエントリーを、キャッシングミスが発生する前にライトバックするので、キャッシングミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。
- [0065] しかも、プロセッサ1は、キャッシング終了属性を設定すべきアドレス範囲として、キャッシュメモリのラインサイズやライン境界とは無関係に任意のアドレスを指定することができる。これにより、プログラマやコンパイラは、ラインサイズ及びライン境界を意識する必要がなくなるので、クリーニングするためのキャッシュアドレス管理をしなくてもよく、プログラムの作成を容易にすることができる。
- [0066] さらに、クリーニング処理において、サブライン単位にダーティなサブランのみをライトバックするので、ライン単位でライトバックする場合と比較してライトバックを高速化することができる。

＜変形例＞

- [0067] なお、本発明のキャッシュメモリは、上記の実施形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。
- [0068] (1)図5において、Cフラグ設定部40は、サイズレジスタ403及び加算器404の代わりにエンドアドレスレジスタを備える構成としてもよい。この場合、エンドアドレスレジスタは、プロセッサ1から書き込まれたエンドアドレスを保持する。

[0069] (2) プロセッサ1が、キャッシング終了属性を付与しながらデータを書き込むストア命令を実行し、制御部38は、さらに、キャッシング終了属性を付与しながらデータを書き込むストア命令を検出する命令検出部と、当該ストア命令による書き込みの際にC=1に設定するフラグ設定部とを備える構成としてもよい。

[0070] (3) 図6(a)(b)(c)に示した各命令は、コンパイラによりプログラム中に挿入してもよい。その際、コンパイラは、例えば配列データの書き込みや、圧縮動画データをデコードする際のブロックデータの書き込み等、これ以上書き込みをしないプログラム位置に、上記各命令を挿入するようにすればよい。

[0071] (4) キャッシュエントリーにCフラグを保持させない構成としてもよい。言い換えれば、図8に示したCフラグ設定処理と図9に示したクリーニング処理とを同時に行う構成としてもよい。この場合、図8におけるS85において、Cフラグをキャッシュエントリーに設定する代わりに、図9におけるループ3(S905～S910)を実行する構成とすればよい。

[0072] (5) 上記実施の形態では、4ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、8ウェイでも16ウェイでもよい。また、上記実施の形態では、セット数が16である例を説明したが、セット数はいくつでもよい。

[0073] (6) 上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式のキャッシュメモリであってもよい。

(実施の形態2)

実施の形態1では、これ以上書き込みをするかしないかを示すCフラグを用いる構成について説明した。本実施の形態では、Cフラグの代わりに、これ以上使用するかしないか(書き込み及び読み出しをするかどうか)を示すW(ウィーク)フラグを用いる構成について説明する。

〈キャッシュメモリの構成〉

[0074] 図10は、本発明の実施の形態2におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図2の構成と比較して、ウェイ31a～31dの代わりにウェイ131a～131dを備える点と、制御部38の代わりに制御部138を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。

[0075] ウェイ131aは、ウェイ31aと比べて、各キャッシュエントリー中に、Cフラグが削除されている点と、Wフラグ及びUフラグが追加されている点が異なる。ウェイ131b～131dも同様である。

[0076] 図11に、キャッシュエントリーのビット構成を示す。1つのキャッシュエントリーは、バリッドフラグV0～V3、21ビットのタグ、128バイトのラインデータ、ウイークフラグW、使用フラグU及びダーティフラグD0～D3を保持する。

[0077] このうち、ウイークフラグWは、プロセッサからのアクセスに関しては、これ以上使用するか否かを意味し、キャッシュメモリにおけるリプレース制御に関しては、他のキャッシュエントリーよりも先に追い出してもよい最弱のリプレース対象を意味する。このように、ウイークフラグWは二つの意味を有することから、クリーニング処理とリプレース処理との2つの処理で参照される。

[0078] 使用フラグUは、そのキャッシュエントリーにアクセスがあったか否かを示し、LRU方式におけるミスヒットによるリプレースに際して4つのウェイのキャッシュエントリー間におけるアクセス順序データの代わりに用いられる。より正確には、使用フラグUの1は、アクセスがあったことを、0はないことを意味する。ただし、1つのセット内の4つウェイの使用フラグが全て1になれば、0にリセットされる。別言すれば、使用フラグUは、アクセスされた時期が古いか新しいか2つの相対的な状態を示す。つまり、使用フラグUが1のキャッシュエントリーは、使用フラグが0のキャッシュエントリーよりも新しくアクセスされたことを意味する。

[0079] 制御部138は、制御部38と比べて、CフラグではなくWフラグを設定する点と、LRU方式におけるアクセス順序情報の代わりに使用フラグUを用いる点とが異なる。
<制御部の構成>

[0080] 図12は、制御部138の構成を示すブロック図である。同図の制御部138は、制御部38と比較して、クリーニング処理部39とCフラグ設定部40との代わりにクリーニング処理部139とWフラグ設定部140とを備える点と、リプレース部41が追加された点とが異なる。

[0081] クリーニング処理部139は、クリーニング処理部39と比べて、CフラグではなくWフラグを参照する点が異なっている。その結果、W=1でかつダーティなキャッシュライ

ンをライトバックする。

[0082] Wフラグ設定部140は、プロセッサ1からのコマンドに応じてキークフラグWを設定する。プロセッサ1は、もはや使用(書き込み及び読み出し)をしないキャッシングエントリーについてキークフラグの設定を指示するコマンドをキャッシングメモリ3に対して発行する。W=1のキャッシングエントリーは、ダーティであればクリーニング処理の対象となり、また、キャッシングミス時には真っ先にリプレース対象となる。

[0083] リプレース部41は、使用フラグUをアクセス順序とする擬似的なLRU方式によりキャッシングミス時にリプレース処理と、キャッシングメモリがアクセスされたとき使用フラグUの更新処理を行う。リプレース処理に際してW=1のキャッシングエントリーは真っ先にリプレース対象として選択される。

<使用フラグUの説明>

[0084] 図13は、リプレース部41による使用フラグの更新例を示す。同図の上段、中断、下段は、ウェイ0～3に跨るセットNを構成する4つのキャッシングエントリーを示している。4つのキャッシングエントリー右端の1又は0は、それぞれ使用フラグの値である。この4つの使用フラグUをU0～U3と記す。

[0085] 同図上段では(U0～U3)=(1, 0, 1, 0)であるので、ウェイ0、2のキャッシングエントリーはアクセスがあったことを、ウェイ1、3のキャッシングエントリーはアクセスがないことを意味する。

[0086] この状態で、メモリアクセスがセットN内のウェイ1のキャッシングエントリーにヒットした場合、同図中段に示すように、(U0～U3)=(1, 1, 1, 0)に更新される。つまり、実線に示すようにウェイ1の使用フラグU1が0から1に更新される。

[0087] さらに、同図中段の状態で、メモリアクセスがセットN内のウェイ3のキャッシングエントリーにヒットした場合、同図下段に示すように、(U0～U3)=(0, 0, 0, 1)に更新される。つまり、実線に示すようにウェイ3の使用フラグU1が0から1に更新される。加えて、破線に示すようにウェイ3以外の使用フラグU0～U2が1から0に更新される。これにより、ウェイ3のキャッシングエントリーが、ウェイ0～2の各キャッシングエントリーよりも新しくアクセスされたことを意味することになる。

[0088] リプレース部41は、キャッシングミス時にW=1のキャッシングエントリーが存在しなけれ

ば、使用フラグに基づいてリプレース対象のキャッシングエントリーを決定してリプレースを行う。例えば、リプレース部41は、図5上段では、ウェイ1とウェイ3の何れかをリプレース対象と決定し、図5中段ではウェイ3をリプレース対象と決定し、図5下段ではウェイ0～2の何れかをリプレース対象と決定する。

＜ウイークフラグWの説明＞

- [0089] 図14(a)ウイークフラグが存在しないと仮定した場合の比較例であり、キャッシングエントリーがリプレースされる様子を示す図である。同図においても、図13と同様にウェイ0～3に跨るセットNを構成する4つのキャッシングエントリーを示している。4つのキャッシングエントリー右端の1又は0は、それぞれ使用フラグの値である。また、データEのみアクセス頻度の低いデータを、データA、B、C、Dはアクセス頻度の高いデータとする。
- [0090] 同図(a)の第1段目の状態で、プロセッサ1がデータEにアクセスすると、キャッシングミスが発生する。このキャッシングミスにより、例えば、 $U=0$ のキャッシングエントリーの中からアクセス頻度の高いデータCのキャッシングエントリーがアクセス頻度の低いデータEにリプレースされ、第2段目の状態となる。
- [0091] 第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシングミスが発生する。このキャッシングミスにより、 $U=0$ のキャッシングエントリーであるアクセス頻度の高いデータDのキャッシングエントリーがアクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。
- [0092] 第3段目の状態で、プロセッサ1がデータDにアクセスすると、キャッシングミスが発生する。このキャッシングミスにより、例えば、アクセス頻度の高いデータCのキャッシングエントリーがアクセス頻度の高いデータDにリプレースされ、第3段目の状態となる。
- [0093] 同様に、第4段目でも、使用頻度の低いデータEはリプレース対象として選択されないで、キャッシングメモリーに残っている。
- [0094] 第5段目の状態で、使用頻度の低いデータEは最も古い($U=0$)であることから、リプレース対象として選択されて、追い出される。
- [0095] このように、擬似LRU方式において(通常のLRU方式においても)、アクセス頻度の低いデータEによって、4ウェイの場合は最悪4回のキャッシングミスを誘発する場合

がある。

[0096] 図14(b)は、リプレース処理におけるウィークフラグWの役割を示す説明図である。

同図(b)の第1段目の状態(同図(a)の第1段目と同じ)で、プロセッサ1がデータEにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、U=0のキャッシュエントリーの中からアクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の低いデータEにリプレースされる。このとき、プロセッサ1は、データEのキャッシュエントリーにウィークフラグWを1に設定するものとする。これにより、次のキャッシュミス時にデータEのキャッシュエントリーが真っ先に追い出され、第2段目の状態となる。

[0097] 第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、W=1のキャッシュエントリーであるアクセス頻度の低いデータEのキャッシュエントリーがリプレース対象として選択され、アクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。

[0098] このように、ウィークフラグWを設けることにより、アクセス頻度の低いデータによるキャッシュミスの誘発を低減することができる。

<Wフラグ設定処理>

[0099] 図15は、Wフラグ設定部140におけるWフラグ設定処理を示すフローチャートである。Wフラグ設定部140の構成は、図5に示したCフラグ設定部と同様であり、また、プロセッサ1からのWフラグ設定コマンドも、図6(a)～(d)と同様である。ただし、プロセッサ1は、これ以上使用しない(書き込みも読み出しあしない)データのアドレス範囲を、Wフラグ設定コマンドとともに指定する。

[0100] 図15は、図8と比べて、ステップS85の代わりにステップS185を有する点が異なっている。これ以外は、図8のCフラグ設定処理と同じであるので、説明を省略する。

ステップS185において、Wフラグ設定部140は、S84によってヒットしたキャッシュエントリーにW=1を設定する。

<クリーニング処理>

[0101] 図16は、クリーニング処理部139におけるクリーニング処理を示すフローチャートである。

同図は、図9と比較して、S903、S904、S911の代わりにS903a、S904a、S911aを有する点が異なっている。S903a、S904a、S911aは、何れもCフラグをWフラグと読み替える点のみが異なり、それ以外は同じ処理内容であるので説明を省略する。

<Uフラグ更新処理>

[0102] 図17は、リプレース部41におけるUフラグ更新処理を示すフローチャートである。

同図では、パリッドフラグが0(無効)であるキャッシュエントリーの使用フラグUは0に初期化されているものとする。

同図において、リプレース部41は、キャッシュヒットしたとき(ステップS61)、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグUを1にセットし(ステップS62)、そのセット内の他のウェイの使用フラグUを読み出し(ステップS63)、読み出した使用フラグUが全て1であるか否かを判定し(ステップS64)、全て1でなければ終了し、全て1であれば他のウェイの全ての使用フラグUを0にリセットする(ステップS65)。

[0103] このようにしてリプレース部41は、図13、図14(a)(b)に示した更新例のように、使用フラグUを更新する。

<リプレース処理>

[0104] 図18は、リプレース部41におけるリプレース処理を示すフローチャートである。同図においてリプレース部41は、メモリアクセスがミスしたとき(ステップS91)、セットインデックスにより選択されたセットにおける、4つウェイの使用フラグU及びウィークフラグWを読み出し(ステップS92)、W=1のウェイが存在するか否かを判定する(ステップS93)。W=1のウェイが存在しないと判定された場合、U=0のウェイを1つ選択する(ステップS94)。このとき、使用フラグUが0になっているウェイが複数存在する場合は、リプレース部41はランダムに1つを選択する。また、W=1のウェイが存在すると判定された場合、Uフラグの値に関わらずW=1のウェイを1つ選択する(ステップS95)。このとき、ウィークフラグWが1になっているウェイが複数存在する場合は、リプレース部41はランダムに1つを選択する。

[0105] さらに、リプレース部41は、当該セットにおける選択されたウェイのキャッシュエントリ

一を対象にリプレースし(ステップS96)、リプレース後に当該キャッシュエントリーの使用フラグUを1に、ウィークフラグWを0初期化する(ステップS97)。なお、このときバリッドフラグV、ダーティフラグDは、それぞれ1、0に初期化される。

- [0106] このように、W=1のウェイが存在しない場合、リプレース対象は、使用フラグUが0のキャッシュエントリーの中から1つ選択される。
- [0107] また、W=1のウェイが存在する場合、リプレース対象は、使用フラグUが0であると1であるとを問わず、W=1のウェイのキャッシュエントリーから1つ選択される。これにより図14(a) (b)に示したように、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。
- [0108] 以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、ウィークフラグW=1のラインを、プロセッサからこれ以上書き込みがなされないラインとして、クリーニングすることにより、キャッシュミス時のライトバックペナルティを低減することができる。
- [0109] また、これ以上使用されないキャッシュエントリーにW=1が設定され、W=1のキャッシュエントリーが真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。
- [0110] また、従来のLRU方式におけるアクセス順序を示すデータの代わりに1ビットの使用フラグを用いる擬似LRU方式を採用することにより、アクセス順序データとして1ビットのフラグでよいので、アクセス順序データのデータ量が少ないと及び更新が簡単であることからハードウェア規模を小さくすることができる。

<変形例>

- [0111] なお、本発明のキャッシュメモリは、上記の実施形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。
- [0112] (1) 使用フラグUを用いる擬似LRUの代わりに、4つのウェイのアクセス順序を示す順序データをキャッシュエントリー毎に保持及び更新して、従来通りのLRU方式でリプレース対象を選択する構成としてもよい。この場合も、W=1のキャッシュエントリーを、アクセス順序に関わらず、真っ先にリプレース対象として選択するようにすればよ

い。

[0113] (2) 実施の形態1に示したCフラグと、本実施の形態におけるWフラグの両方設ける構成としてもよい。この場合、Cフラグはクリーニング処理で用いて、Wフラグはリプレース対象の選択のみに用いるようにすればよい。こうすれば、Cフラグによって、読み出しされる可能性があつてかつこれ以上書き込みされないキャッシングエントリーは、クリーニングされた後もキャッシングメモリに残すことができる。これ以上読み出しも書き込みもなれないキャッシングエントリーは、クリーニングされかつリプレース対象として真っ先に選択される。CフラグとWフラグの併用によって、きめ細かい制御を行うことができる。

[0114] (3) 上記実施の形態では、サブラインのサイズをラインのサイズの1/4としているが、1/2、1/8、1/16等他のサイズでもよい。その場合、各キャッシングエントリーは、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。

産業上の利用可能性

[0115] 本発明は、メモリアクセスを高速化するためのキャッシングメモリに適しており、例えば、オンチップキャッシングメモリ、オフチップキャッシングメモリ、データキャッシングメモリ、命令キャッシングメモリ等に適している。

請求の範囲

[1] ラインデータを保持するキャッシュエントリー毎に、当該キャッシュエントリーにキャッシングを終了してよいかどうかを示すキャッシング終了属性を付与する付与手段と、キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択する選択手段と、
選択されたキャッシュエントリーのラインデータをキャッシングの発生と無関係にメモリへライトバックするライトバック手段と
を備えることを特徴とするキャッシングメモリ。

[2] 前記付与手段は、
プロセッサから指定されたアドレス範囲を保持する保持手段と、
保持手段に保持されたアドレス範囲に属するラインデータを保持するキャッシュエントリーを探索する探索手段と、
探索されたキャッシュエントリーに、キャッシングを終了してよいことを示すキャッシング終了属性を設定する設定手段と
を備えることを特徴とする請求の範囲第1項に記載のキャッシングメモリ。

[3] 前記探索手段は、
保持手段に保持されたアドレス範囲の先頭アドレスがラインデータの途中を指す場合、当該先頭アドレスを、前記アドレス範囲に含まれる先頭のラインを指すスタートラインアドレスに変換する第1変換手段と、
保持手段に保持されたアドレス範囲の末尾アドレスがラインデータの途中を指す場合、当該末尾アドレスを、前記アドレス範囲に含まれる末尾のラインを指すエンドラインアドレスに変換する第2変換手段と、
前記スタートラインアドレスからエンドラインアドレスまでの各ラインアドレスに対応するデータを保持するキャッシュエントリーがあるか否かを判定する判定手段と
を備えることを特徴とする請求の範囲第2項に記載のキャッシングメモリ。

[4] 前記キャッシングメモリは、さらに、キャッシング発生時に、キャッシングを終了してよいことを示すキャッシング終了属性が付されたキャッシュエントリーをリプレース対象と

して選択するリプレース手段を備える

ことを特徴とする請求の範囲第2項に記載のキャッシュメモリ。

[5] 前記付与手段は、

プロセッサにおいてキャッシングを終了してよいことを示すキャッシング終了属性の付与とデータの書き込みとを命令内容とするストア命令が実行されたことを検出する命令検出手段と、

検出された当該命令によって書き込みがなされたキャッシュエントリーに、キャッシング終了属性を設定する設定手段と

を備えることを特徴とする請求の範囲第1項に記載のキャッシュメモリ。

[6] 前記ライトバック手段は、メモリバスが空きサイクルになっているとき、キャッシュエントリーのデータをメモリへライトバックする

ことを特徴とする請求の範囲第1項に記載のキャッシュメモリ。

[7] 前記キャッシュエントリーは、1ラインを構成する複数のサブライン毎にダーティフラグを有し、

前記ライトバック手段は、選択手段に選択されたキャッシュエントリーについてダーティなサブラインのみをライトバックする

ことを特徴とする請求の範囲第1項に記載のキャッシュメモリ。

[8] キャッシュメモリの制御方法であって、

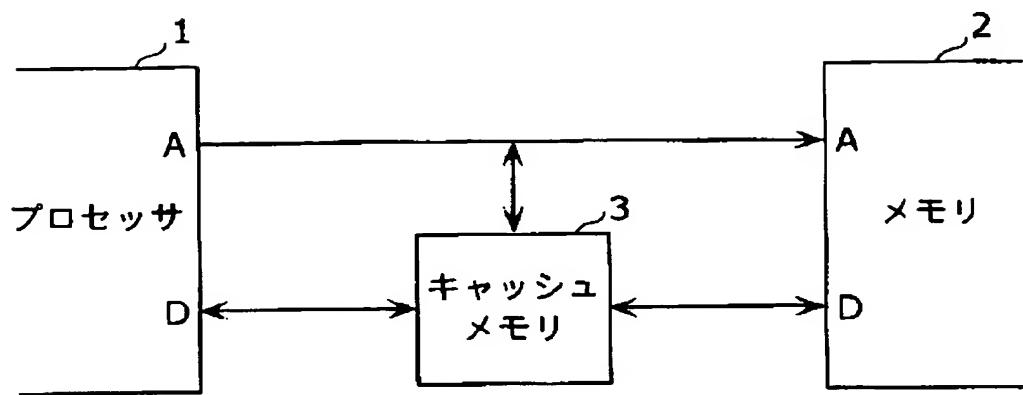
ラインデータを保持するキャッシュエントリー毎に、キャッシングを終了していかどうかを示すキャッシング終了属性を付与する付与ステップと、

キャッシングを終了してよいことを示すキャッシング終了属性が付与され、かつ書き込みされたことを示すダーティフラグがセットされているキャッシュエントリーを選択する選択ステップと、

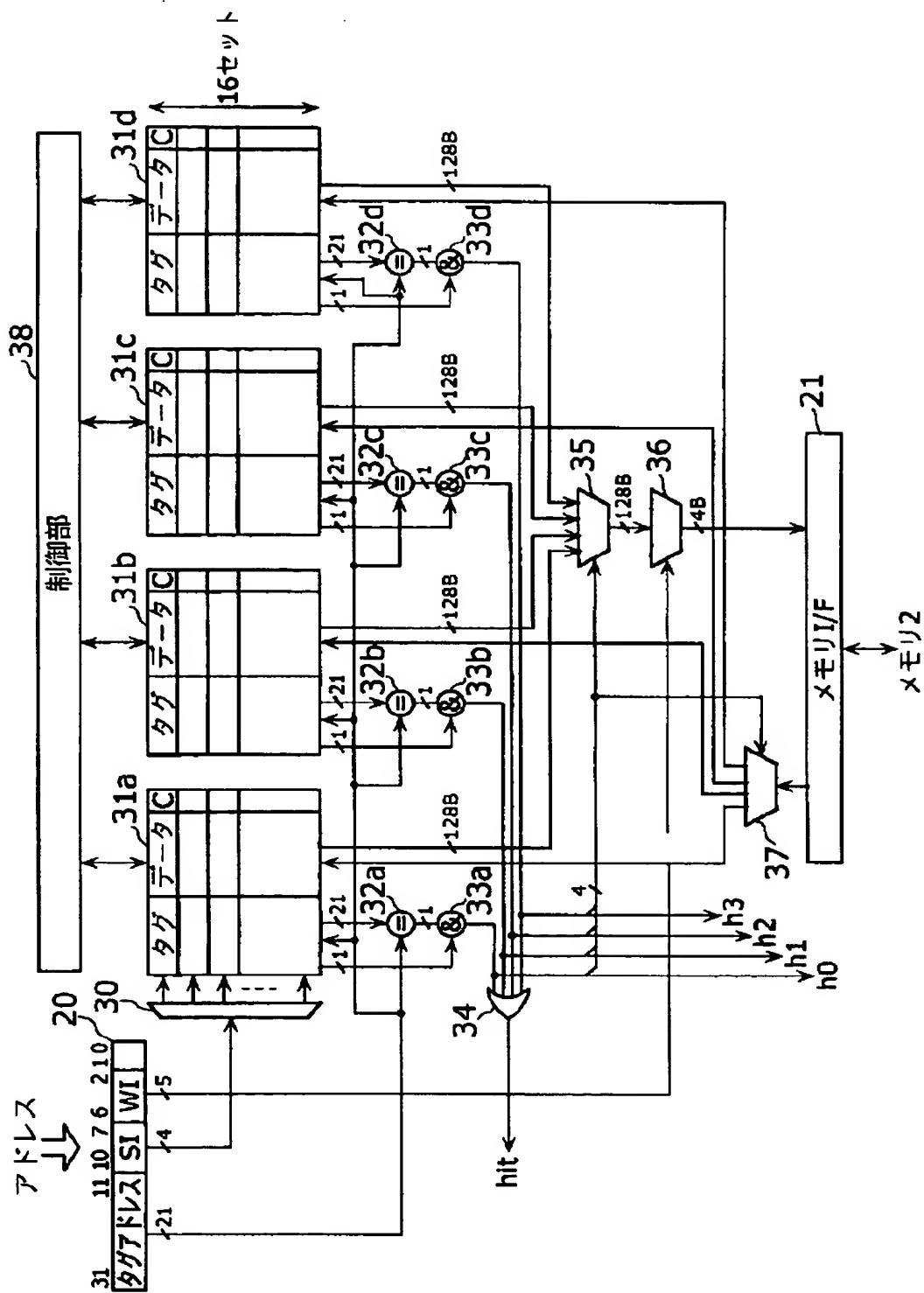
選択されたキャッシュエントリーのラインデータをキャッシュミスの発生と無関係にメモリへライトバックするライトバックステップと

を有することを特徴とする制御方法。

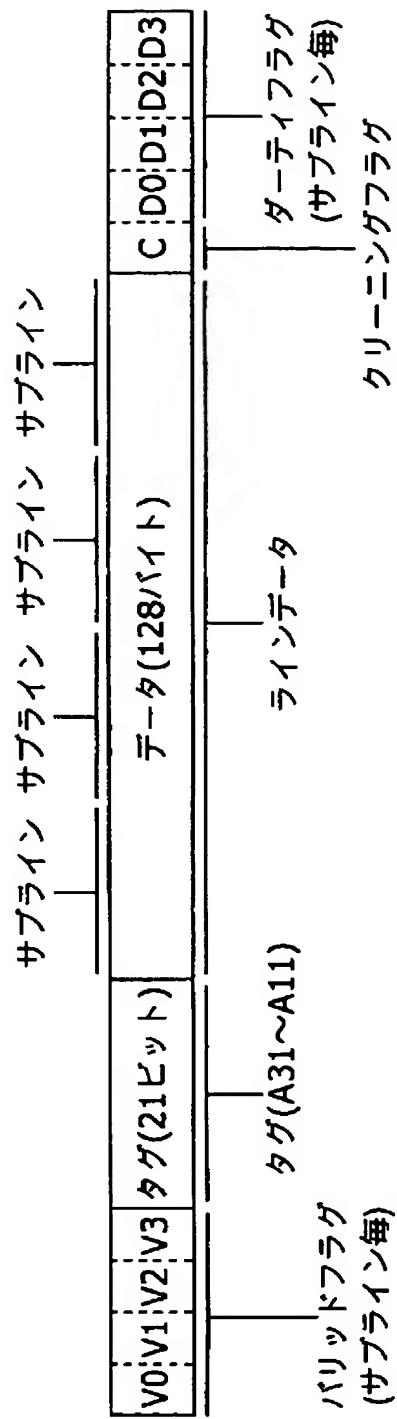
[図1]



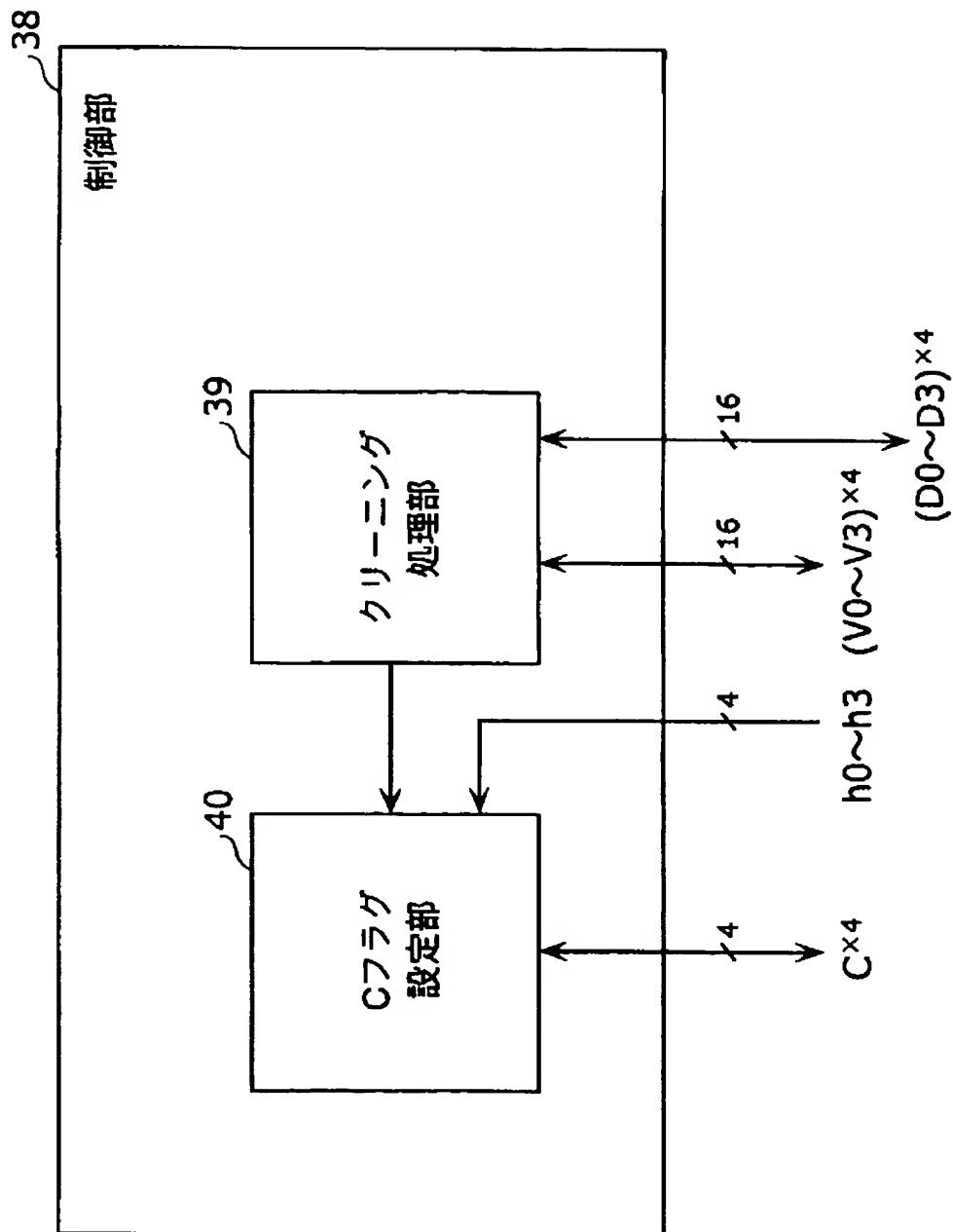
[図2]



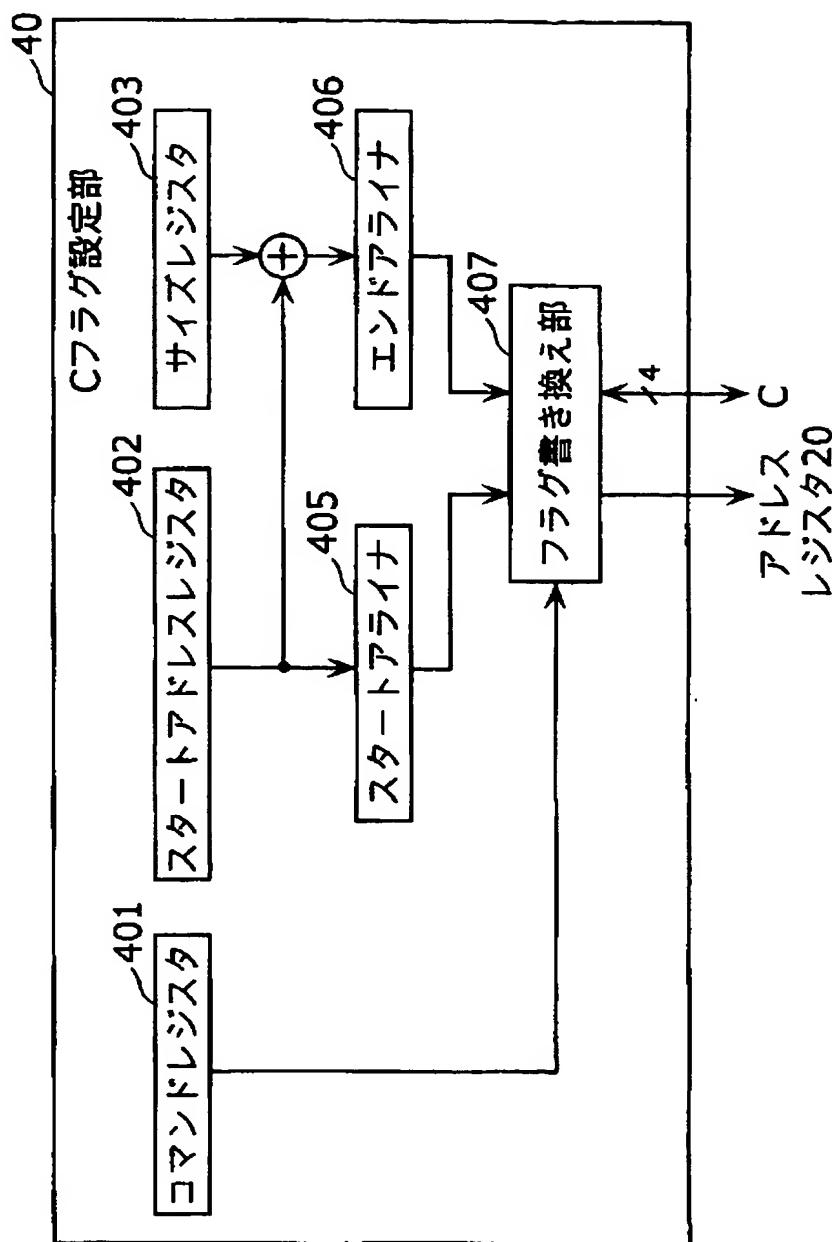
[図3]



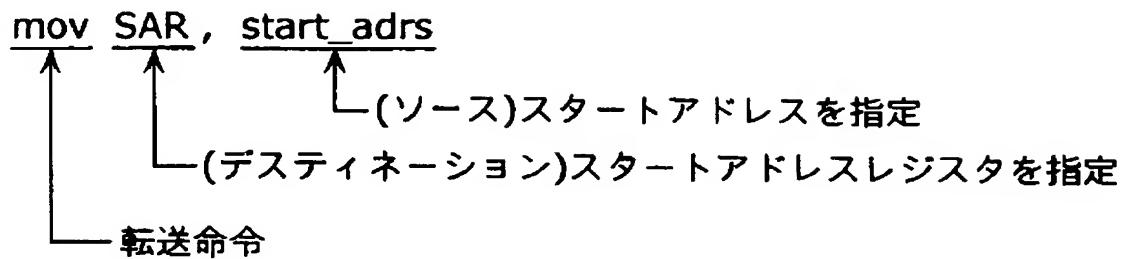
[図4]



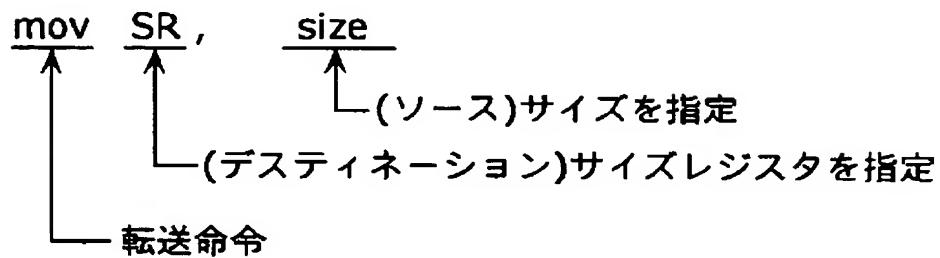
[図5]



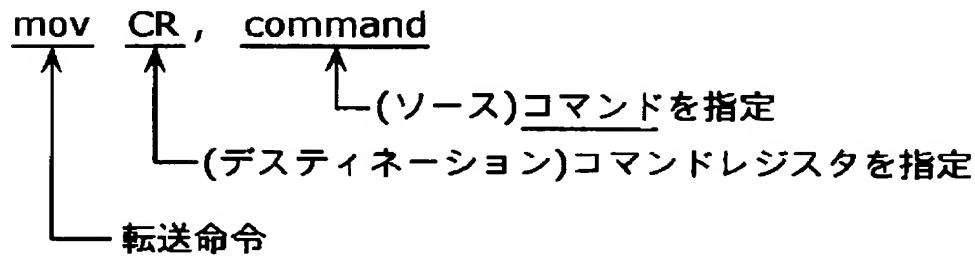
[ 6A]



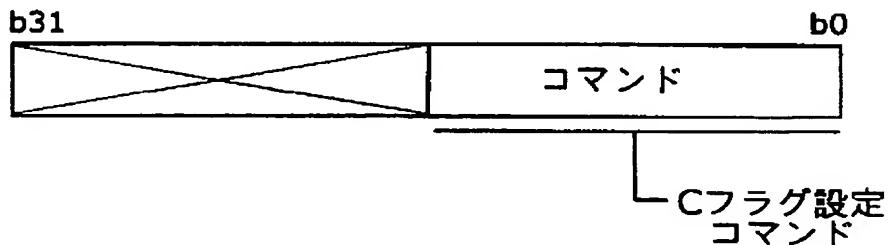
[図6B]



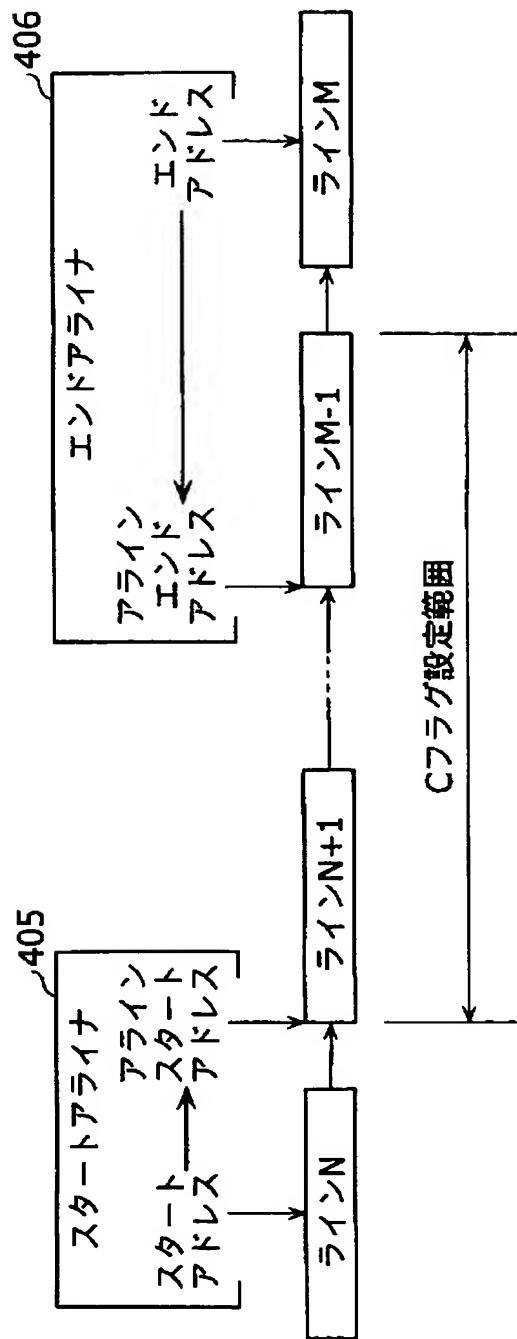
[図6C]



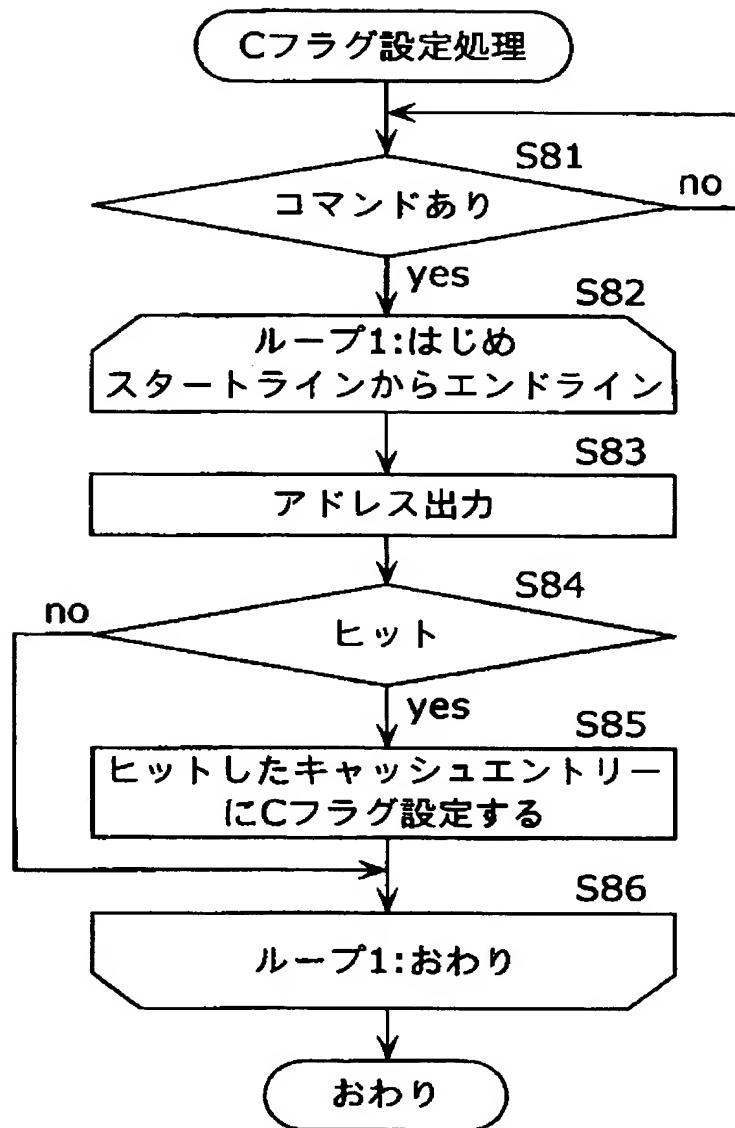
[6D]



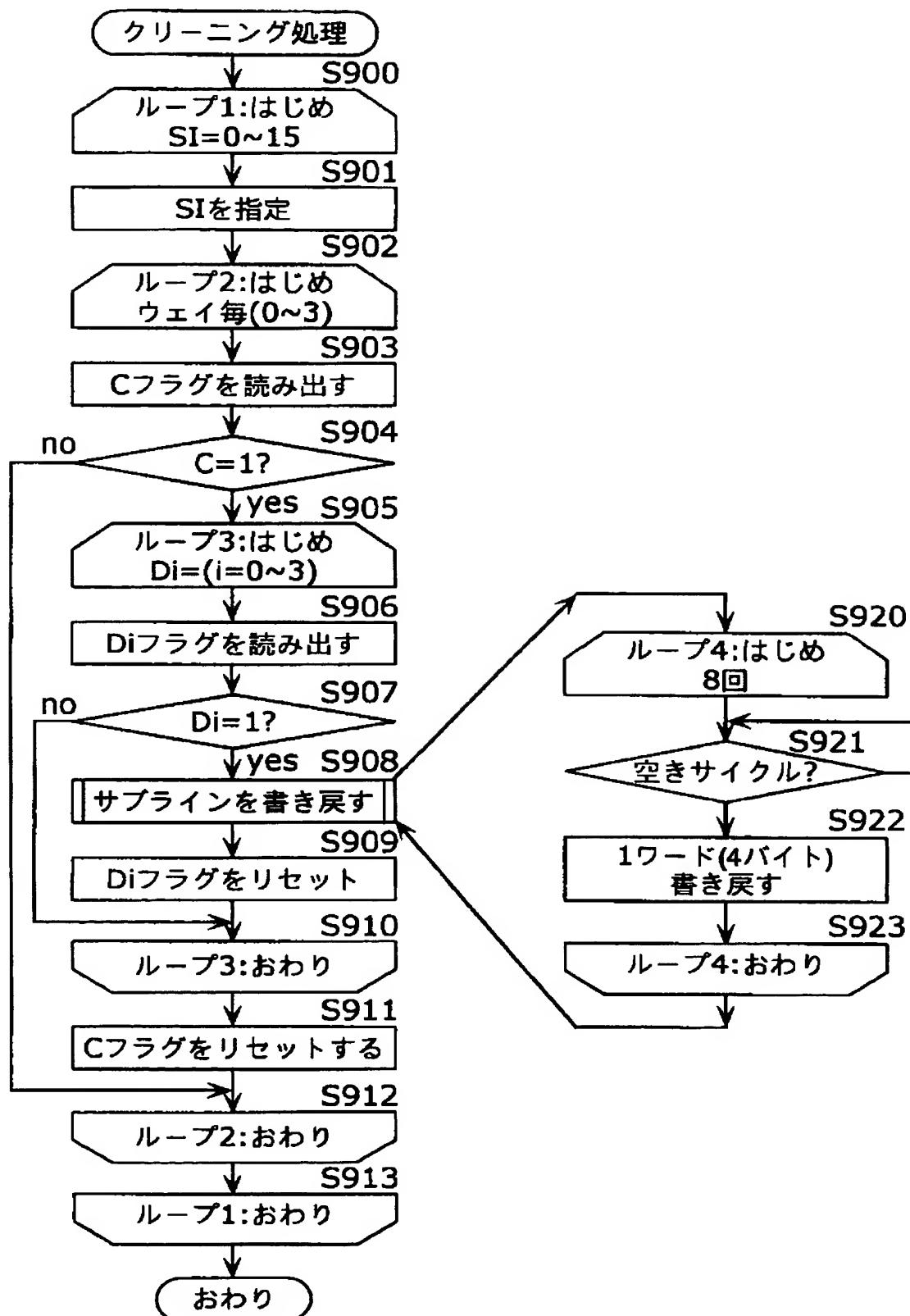
[図7]



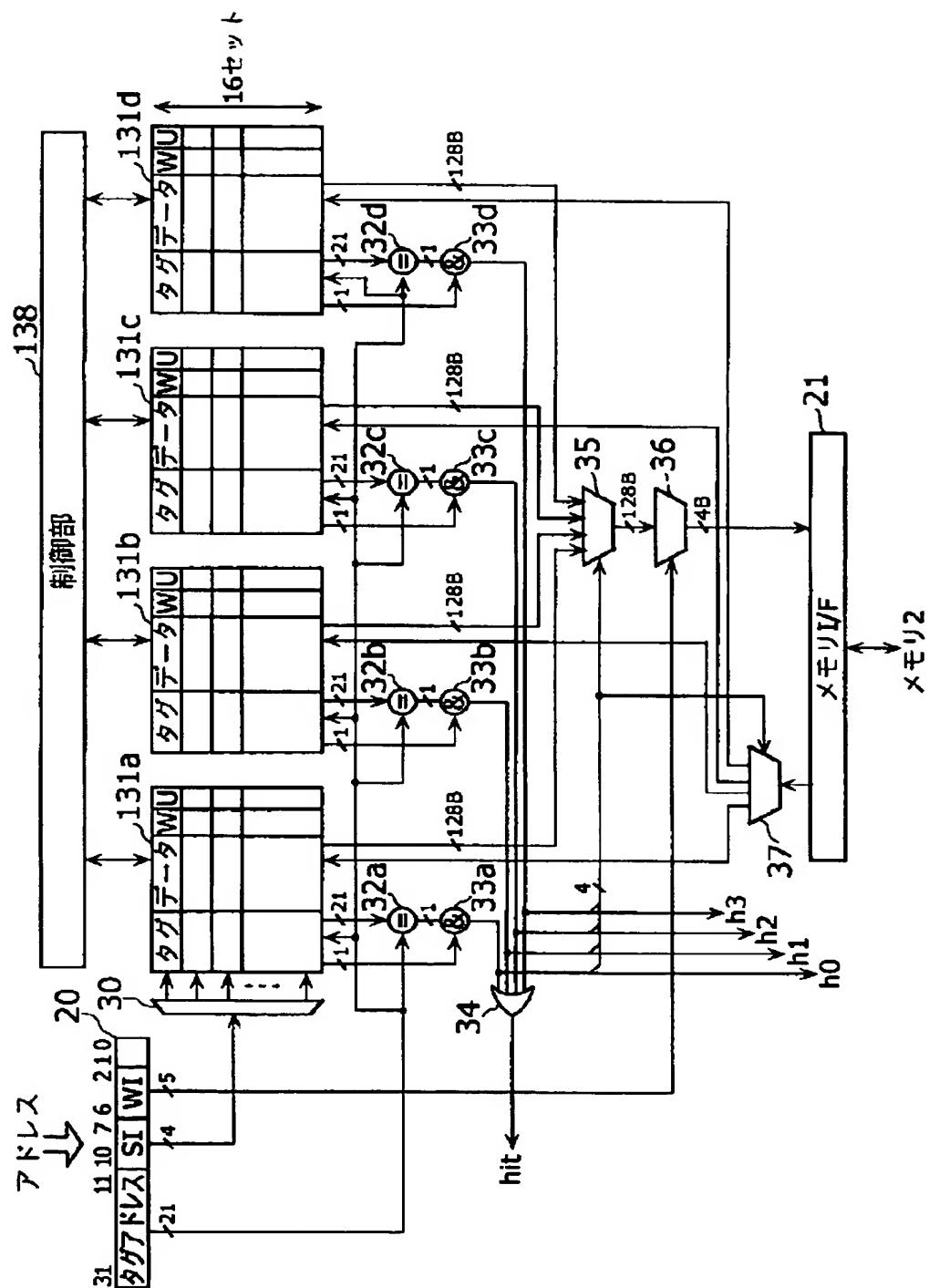
[図8]



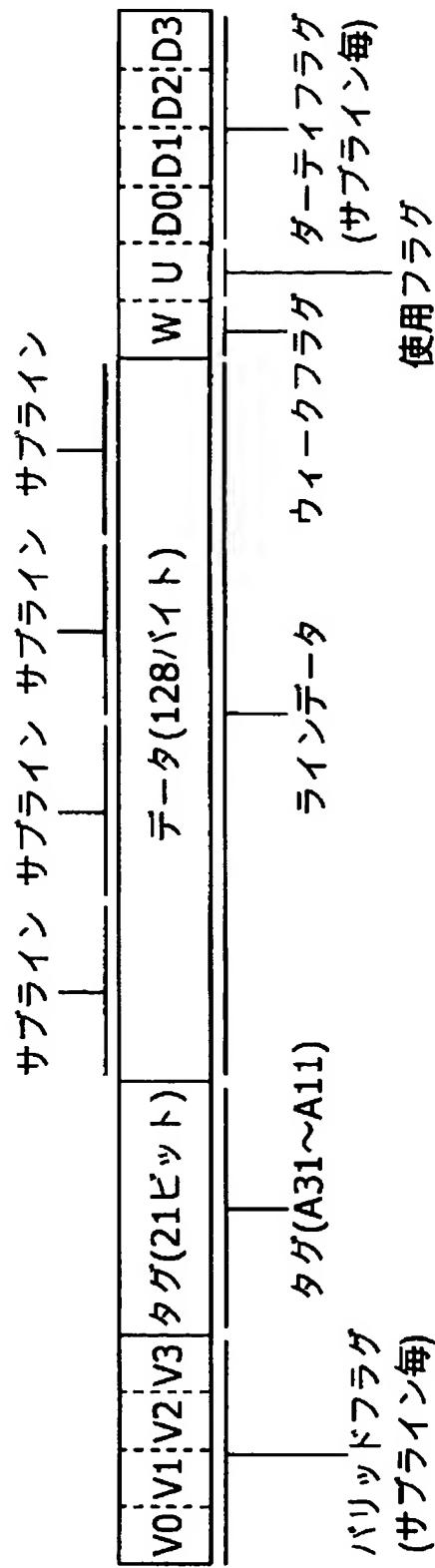
[図9]



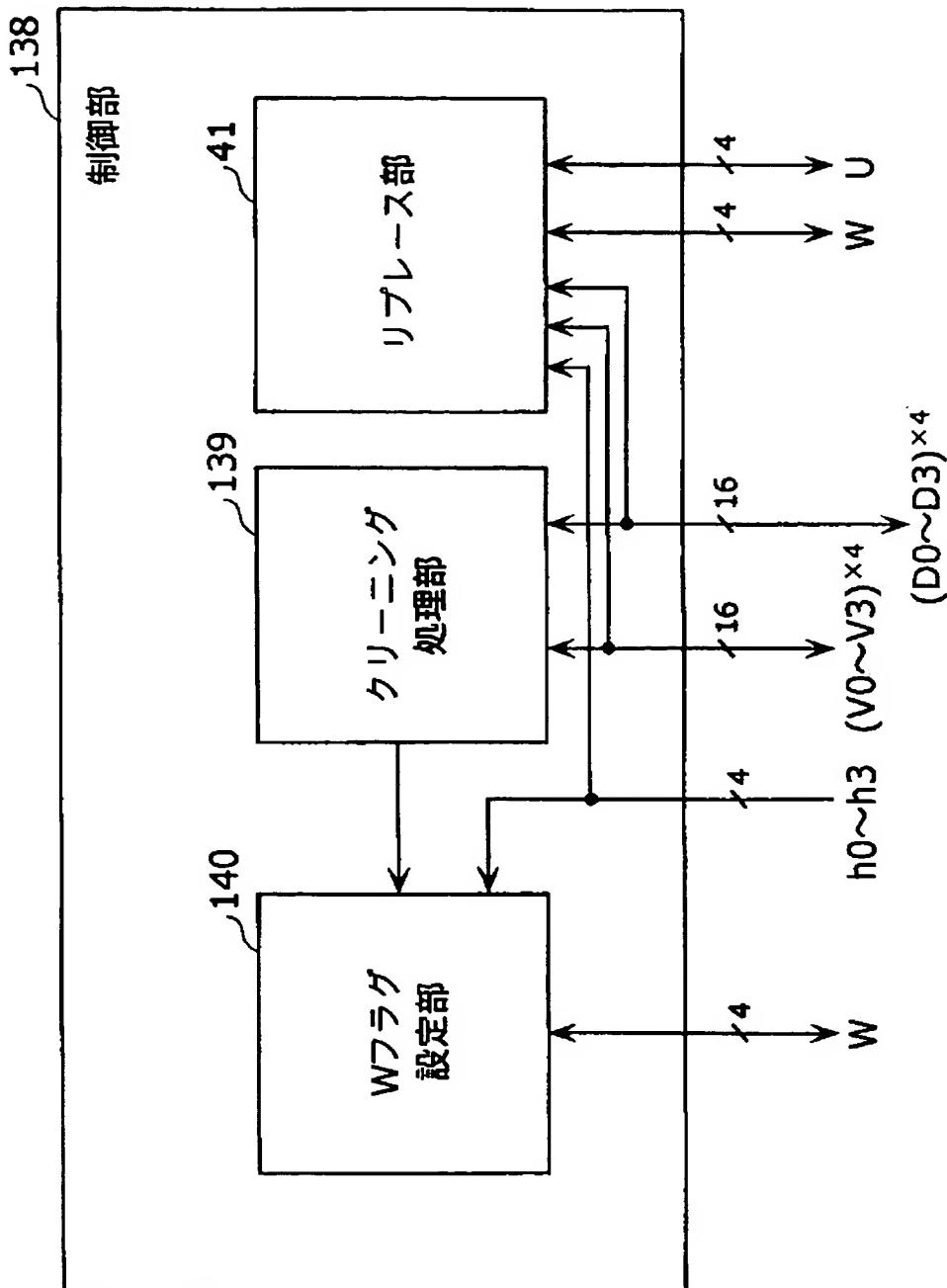
[図10]



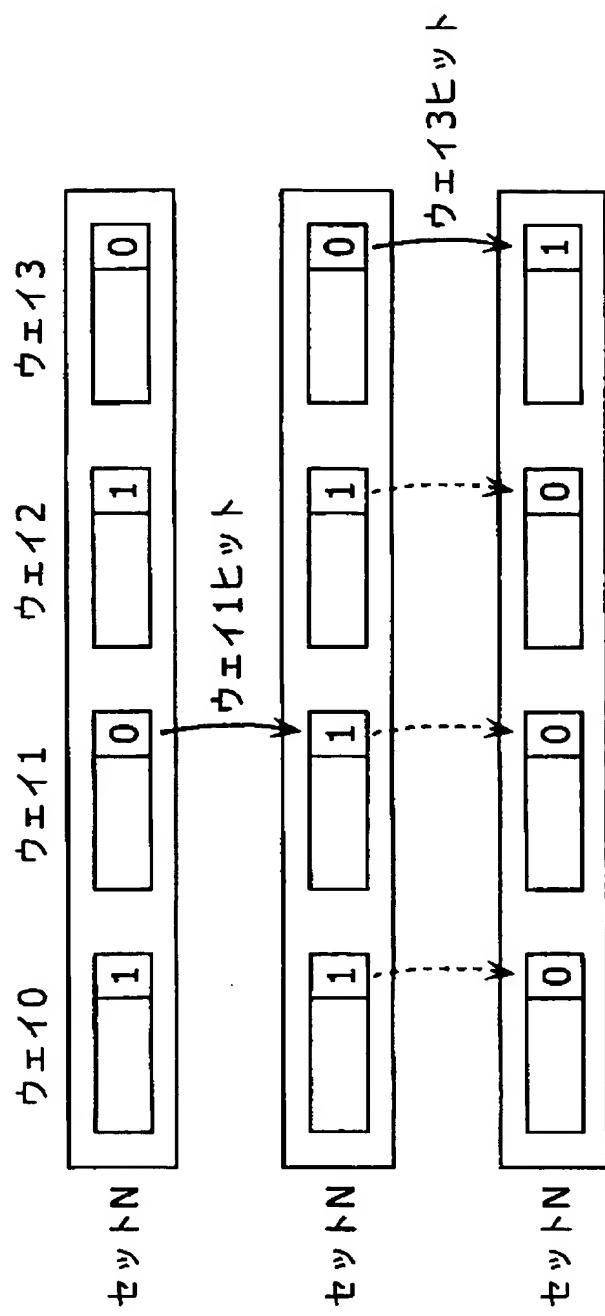
[図11]



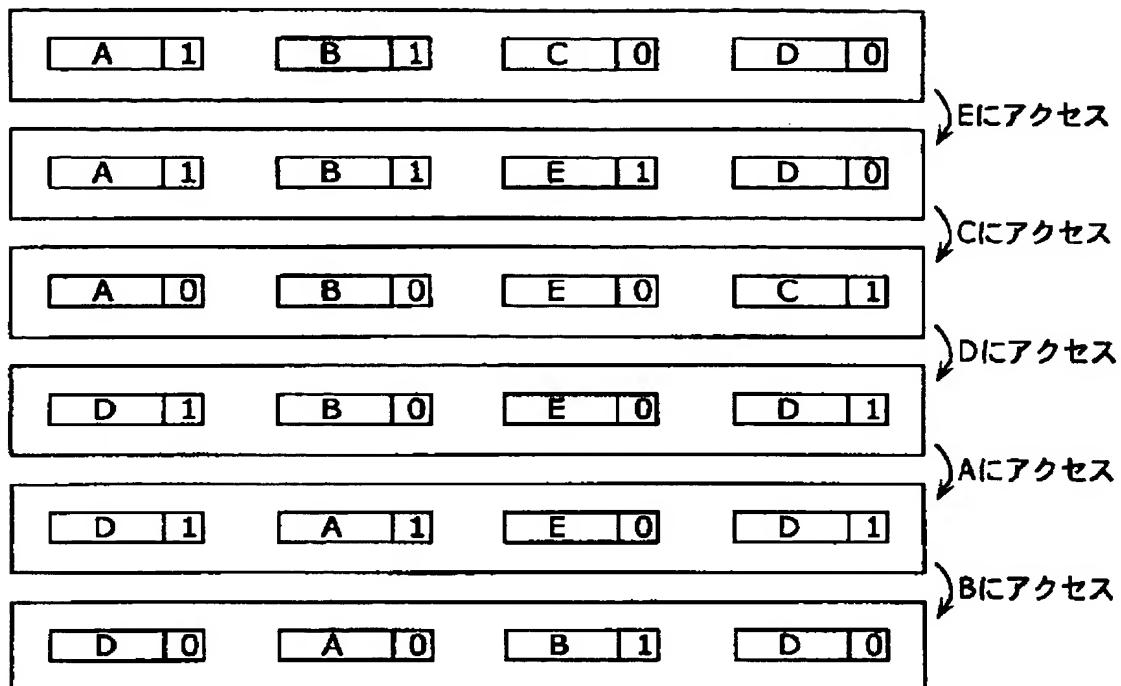
[図12]



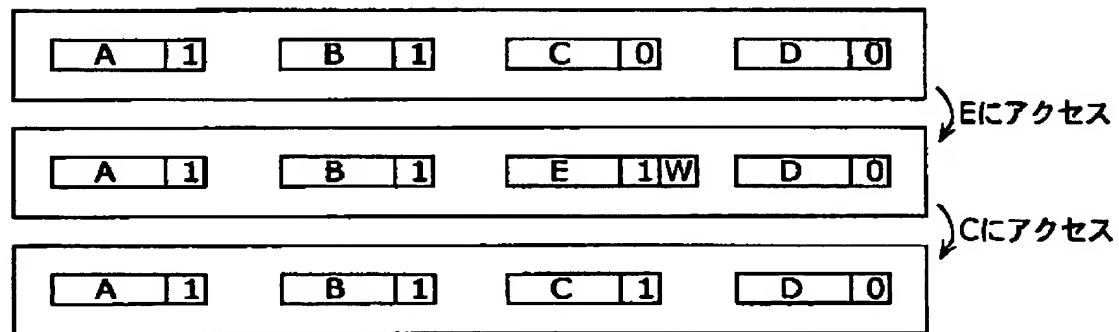
[図13]



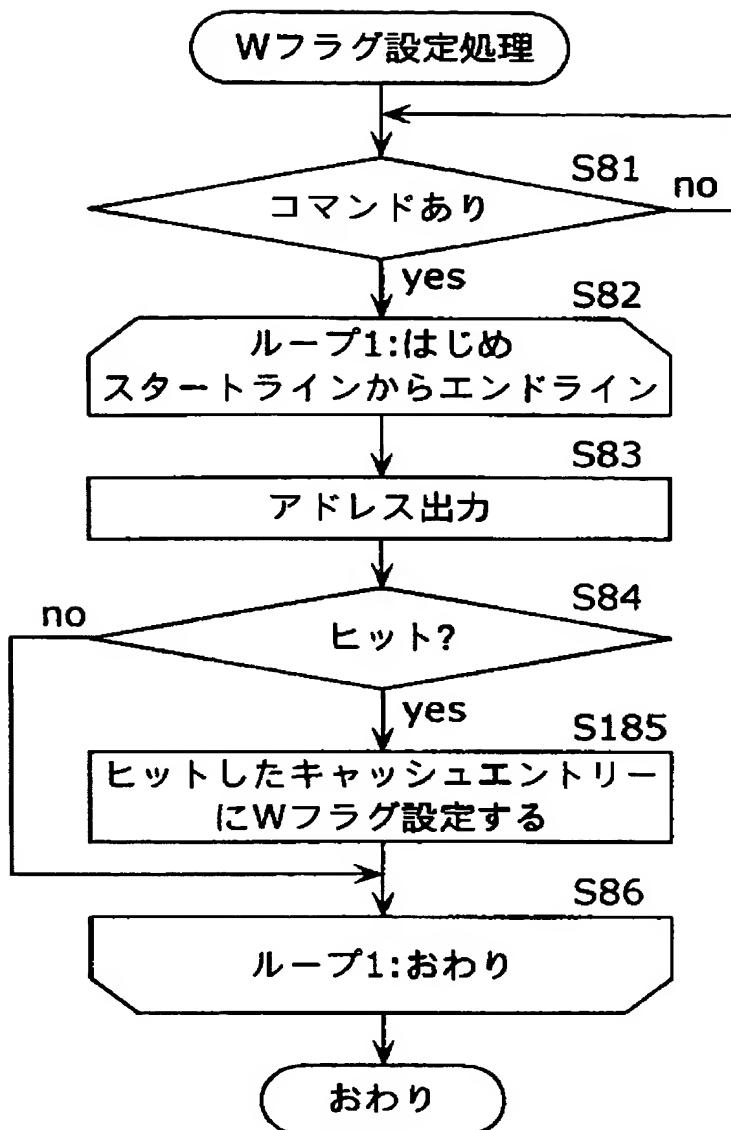
[図14A]



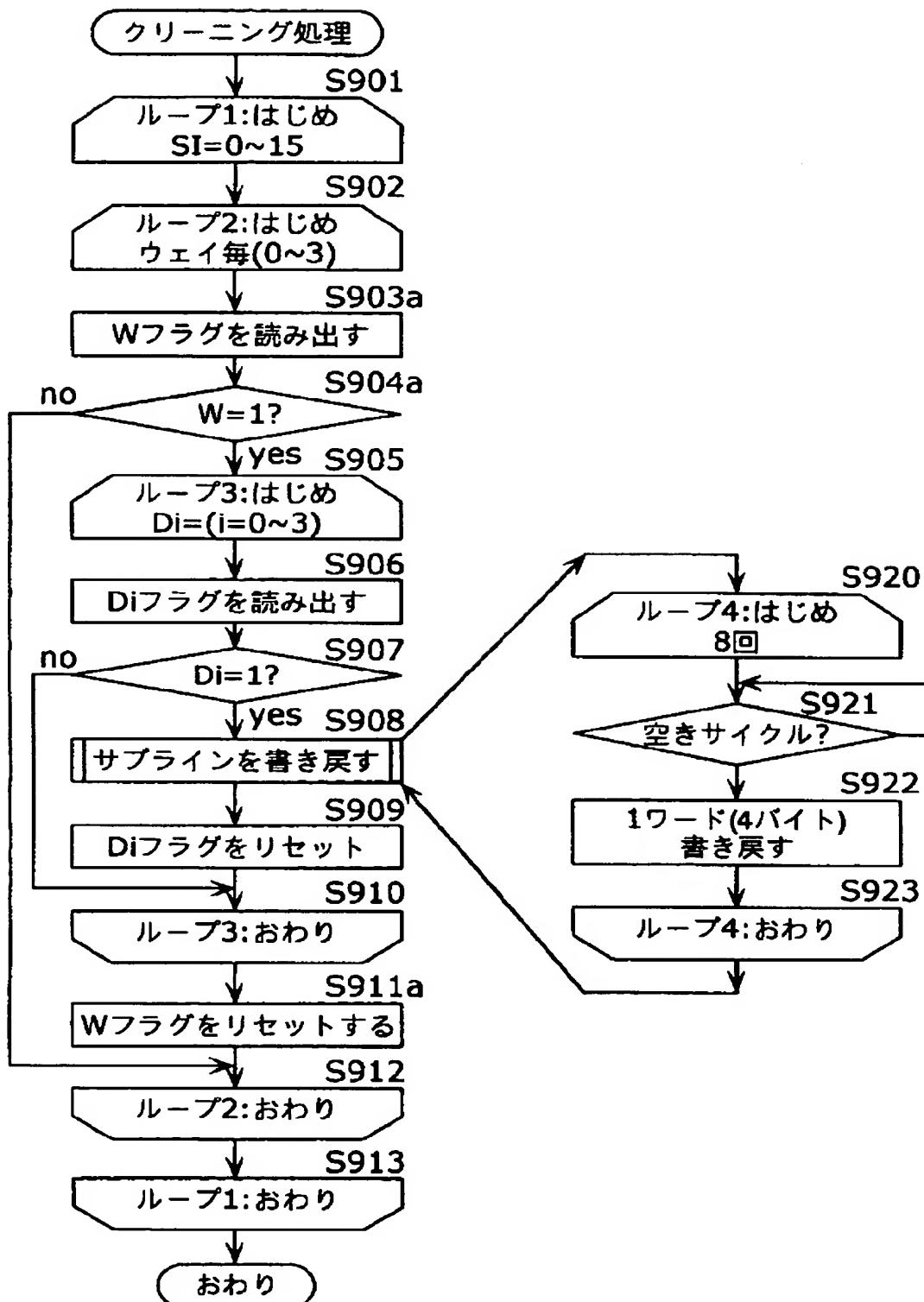
[図14B]



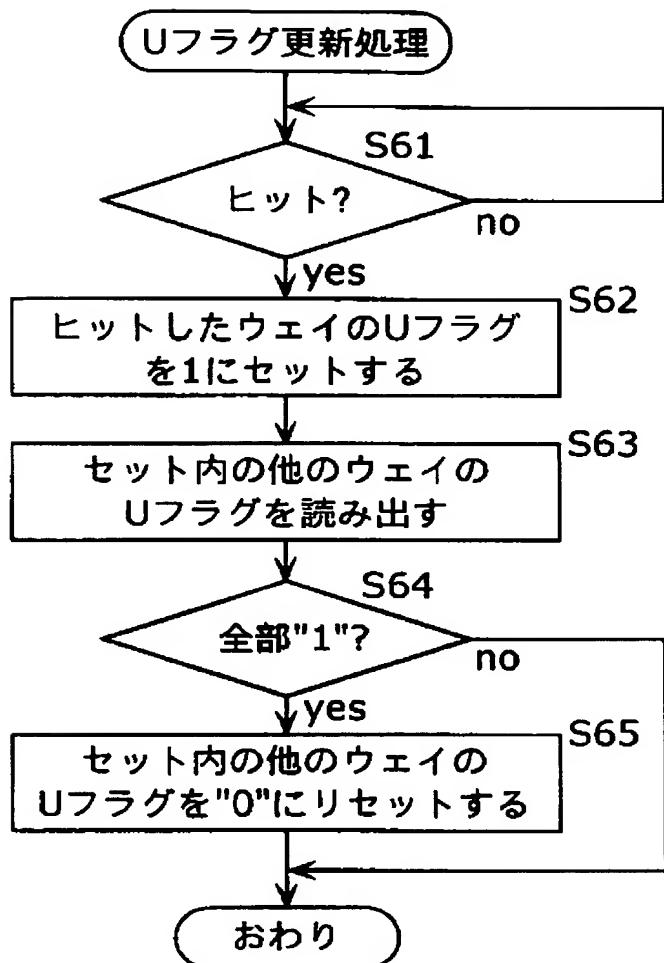
[図15]



[図16]



[図17]



[図18]

